Table of Contents

PBS on Pleiades

Overview

Overview

On Pleiades, PBS (version 10.4) is used to manage batch jobs that run on the compute nodes (3 different processor types, 9,984 nodes and 91,136 cores in total). PBS features that are common to all NAS systems are described in other articles. Read the following articles for Pleiades-specific PBS information:

- queue structure

- resource request examples
- default variables set by PBS
- sample PBS scripts

Queue Structure

Users should be aware of the PBS queue structure. To find the maximum and default NCPUS (number of CPUs), the maximum and default wall time, the priority of the queue, and whether the queue is disabled or stopped, use the command:

`%qstat -Q`
This command also provides statistics of jobs in the states of queued (Q), held (H), running (R), or exiting (E).

Note that the queue structure will change from time to time. Below is a snapshot of the output from this command on June 16, 2011.

```
%qstat -Q
Queue    Ncpus/      Time/          State counts
name       max/def    max/def   jm T/_Q/H/W/__R/E/B pr  Info
======== =====/=== ======/===== == ================ === ========
normal      --/  8  08:00/01:00 -- 0/20/4/0/_60/0/0   0
debug     1025/  8  02:00/00:30 -- 0/_3/0/0/__4/0/0  15
low         --/  8  04:00/00:30 -- 0/_0/0/0/__0/0/0 -10
long      8192/  8 120:00/01:00 -- 0/_8/1/0/206/0/0   0
route       --/  8     --/   -- -- 0/_0/0/0/__0/0/0   0
idle        --/ --     --/   -- -- 0/_0/0/0/__0/0/0   0 disabled
alphatst    --/ -- 120:00/01:00 -- 0/_0/0/0/__0/0/0   0
ded_time    --/ --     --/01:00 -- 0/_0/0/0/__0/0/0   0
devel     4800/  1  02:00/   -- -- 0/_1/0/0/__5/0/0   0
wide        --/ -- 120:00/01:00 -- 0/_1/0/0/__0/0/0  45 disabled
testing     --/ --     --/00:30 -- 0/_0/0/0/__0/0/0   0
somd_spl    --/  8 240:00/01:00 -- 0/_0/0/0/__2/0/0  25
armd_spl  4900/  8 120:00/01:00 10 0/_0/0/0/__0/0/0  15
normal_N    --/  8 120:00/01:00 -- 0/_5/0/0/_10/0/0   0
rtf         --/  8     --/01:00 -- 0/_0/0/0/__0/0/0  65
dpr         --/  8     --/00:10 -- 0/_0/0/0/__0/0/0   0
normal_W    --/  8 120:00/01:00 -- 0/60/5/0/_18/0/0   0
S1848368   744/  1  04:00/   -- -- 0/_0/0/0/__0/0/0   0
kepler      --/  8 120:00/01:00 -- 0/12/0/0/_33/0/0  20
diags       --/ -- 120:00/01:00 -- 0/_0/0/0/__0/0/0   0
```

The *devel* queue on Pleiades is served by pbspl3 (a non-default PBS server). The *devel* queue requires pbspl3 to be included in the syntax for qsub, qstat, and qdel. For more information, read the article Pleiades devel Queue.
To view more information about each queue, use:

`%qstat -fQ queue_name`
In the output of this command, you will find additional information such as:

acl_groups
: Lists all GIDs that are allowed to run in the queue.
For the *normal, debug, long, low* and *wide* queues, all GIDs should be included.
Special queues, such as *esmd_spl, armd_spl, somd_spl, clv_spl*, etc., typically allow

a few GIDs only.

default_chunk.model
: Specifies the default processor model, if you do not specify the processor model yourself.
All queues, except *normal_N* and *normal_W*, default to using nodes with Harpertown model processors.

resources_min.ncpus
: If defined, specifies the minimum NCPUs required for the queue.
The *wide* queue requires using a minimum of 1024 CPUs.

max_user_run
: If defined, specifies the maximum number of jobs each user is allowed to run in the queue.
For the *normal* queue, it is set at 10. For the *debug* queue, it is set at 2.

The *normal_N* and *normal_W* queues will be removed in the near future. To request using the Nehalem-EP or Westmere nodes, use "model=neh" or "model=wes" attribute in your *resource_list*. To explicitly request Harpertown nodes, use "model=har". You can apply the model attribute to any queue.

For example:

```
#PBS -q long
#PBS -l select=1:ncpus=12:model=wes
```

Mission Shares Policy on Pleiades

DRAFT

This article is being reviewed for completeness and technical accuracy.

Mission Directorate shares have been implemented on Pleiades since Feb. 10, 2009. Implementing shares guarantees that each Mission Directorate gets its fair share of resources.

The share to which a job is assigned is based on the GID used by the job. Once all the cores within a Mission Directorate's share have been assigned, other jobs assigned to that share must wait, even if cores are available in a different Mission Directorate's share, with the following exception:

When a Mission Directorate is not using all of its cores, other Mission Directorates can borrow those cores, but only for jobs that will finish within 4 hours. When part of the resource is unavailable, the total number of cores decreases, and each Mission Directorate loses a proportionate number of cores.

You can display the share distribution by adding the "-W shares=-" option to the qstat command. For example:

```
%qstat -W shares=-

Group   Share% Use%  Share Exempt   Use  Avail Borrowed Ratio Waiting
------- ------ ---- ------ ------ ----- ------ -------- ----- -------
Overall    100    0 159748      0   960 158788        0  0.01     960
 ARMD       24   18  38109      0 29680   8429        0  0.78   22512
 HEOMD      23   21  36521      0 34312   2209        0  0.94   28416
 SMD        51   50  80981      0 80968     13        0  1.00  113920
 NAS         2    0   3175      0     0   3175        0  0.00   20240
```

Mission shares are calculated by combining the mission's HECC share of the shared assets combined with the mission-specific assets. The mission shares on Oct 3, 2011 are shown in the second column of the above display. The amount of resources used and borrowed by each mission and resources each mission is waiting for are also displayed.

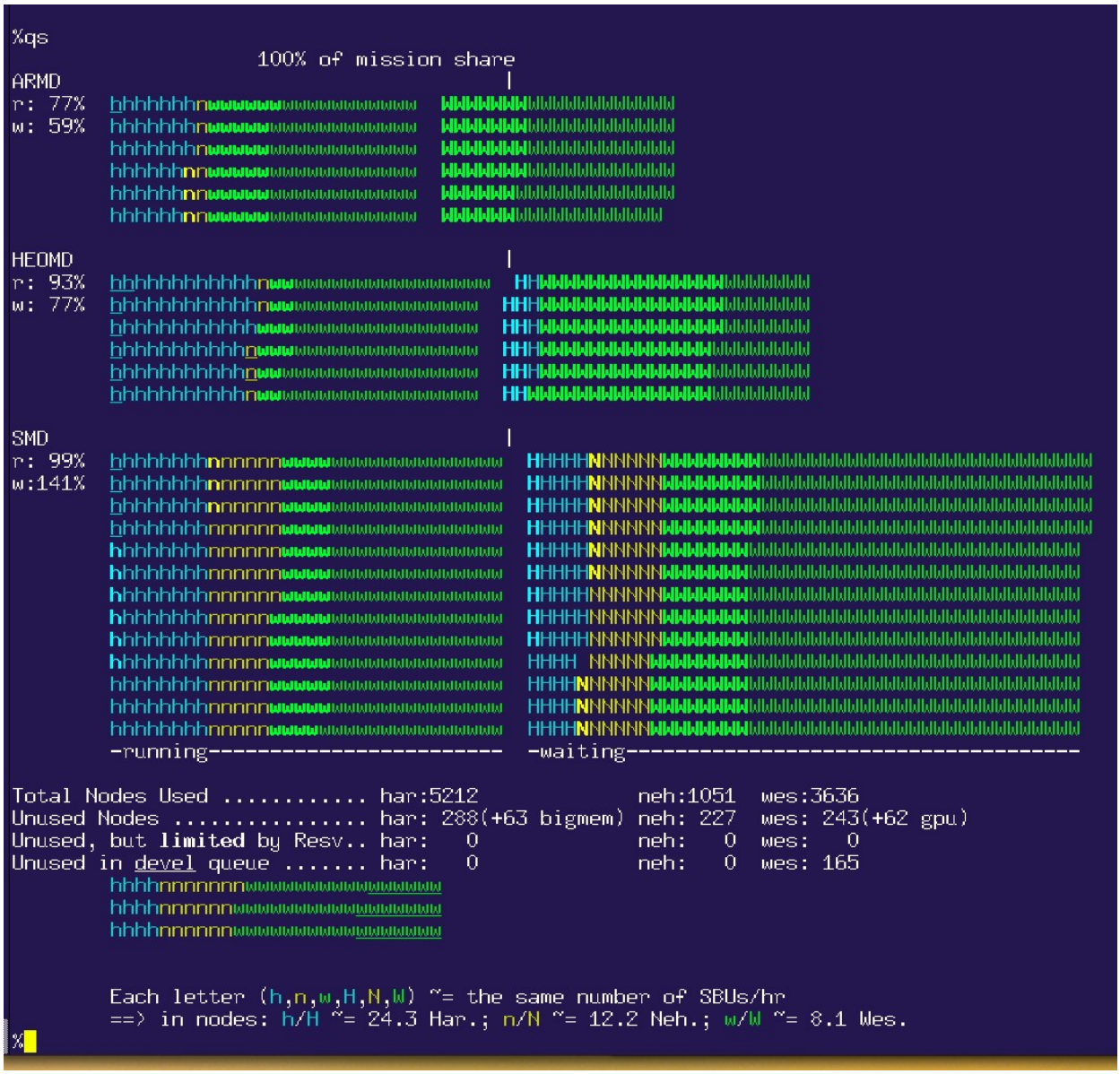
An in-house utility, *qs*, provide similar information with details that break the resources into the Harpertown, Nehalem-EP and Westmere-EP processor types and is available at /u/scicon/tools/bin/qs.

The -h option of qs provides instructions on how to use it:

```
% /u/scicon/tools/bin/qs -h
```

```
usage: qs [-u] [-n N] [-b] [-p] [-d] [-r] [-f M,N] [-q N] [-t] [-v] [-h] [--file f]
          -u       : show used resources only; don't show queued jobs
          -n N     : show time remaining before N nodes are free
          -b       : order segments in bars to help understand borrowing
          -p       : plain output: i.e. no colors or highlights
          -d       : darker colored resource bars (for a light background)
          -r       : use Reverse video for displaying resource bars
          -f M,N   : highlight nodes for jobs that finish in <= M minutes
                     and <= N minutes [default M=60,N=240]
                     (0 turns off highlighting)
          -q N     : highlight nodes for jobs queued in last N minutes [3]
                     (0 turns off highlighting)
          -t       : show time remaining & nodes used for each running job
          --file f : reserved for debugging
          -v       : (verbose) provide explanation of display elements
          -h       : provide this message
```

Here is a sample output file of *qs*:


%qs
100% of mission share
ARMD
r: 77%
w: 59%
HEOMD
r: 93%
w: 77%
SMD
r: 99%
w:141%
-running-
-waiting-
Total Nodes Used har:5212 neh:1051 wes:3636
Unused Nodes ..... har: 288(+63 bigmem) neh: 227 wes: 243(+62 gpu)
Unused, but limited by Resv.. har: 0 neh: 0 wes: 0
Unused in devel queue ..... har: 0 neh: 0 wes: 165
Each letter (h,n,w,H,N,W) ~= the same number of SBUs/hr
==> in nodes: h/H ~= 24.3 Har.; n/N ~= 12.2 Neh.; w/W ~= 8.1 Wes.
%

Resources Request Examples

Since Pleiades consists of three different processor types, Harpertown, Nehalem-EP and Westmere-EP, keep the following in mind when requesting PBS resources for your job:

- Starting May 1, 2011, charging on the usage of the three Pleiades processor types is based on a common Standard Billing Unit which is on a per-node basis. The SBU rate for each of the Pleiades processor type is:

```
Processor Type        SBU Rate (per node)
Westmere-EP           1    (12 cores in a node)
Nehalem-EP            0.8  (8 cores in a node)
Harpertown            0.45 (8 cores in a node)
```

  The actural amount of memory per node through PBS is slightly less, 7.6 GB/node for Harpertown and 22.5 GB/node for Nehalem-EP and Westmere-EP.

  Use the "model=[har,neh,wes]" attribute to request the processor type(s) for your job. If the processor type is not specified in user's PBS resource list, the job is routed to use the default processor type, Harpertown.

  Example 1:

  Here are some examples of requesting certain processor models for a 128-process job:

```
#PBS -l select=16:ncpus=8:model=har
# to run all 8 cores on each of 16 Harpertown nodes

#PBS -l select=32:ncpus=4:model=har
# to run on only 4 cores on each of 32 Harpertown nodes
# (note: will be charged for 32 nodes = 256 cores)

#PBS -l select=16:ncpus=8:model=neh
# to run all 8 cores on each of 16 Nehalem-EP nodes

#PBS -l select=11:ncpus=12:model=wes
# to run all 12 cores on each of 11 Westmere-EP nodes
# (4 cores in 11th node will go unused)
```

  Note that you can specify both the queue type (-q normal, debug, long, wide, low) and the processor type (-l model=har,neh,wes). For example:

```
#PBS -q normal
#PBS -l select=16:ncpus=8:model=neh
```

  If your application can run on any of the three processor types, you may want to submit your job to a processor type that has more unoccupied nodes by other

running jobs. This can possibly reduce the wait time of your job. The script *node_stats.sh* provides information about the total, used and free nodes for each processor type. For example:

```
%/u/scicon/tools/bin/node_stats.sh


 Pleiades Nodes Total: 9394
 Pleiades Nodes Used : 8128
 Pleiades Nodes Free : 1266

 Harpertown Total: 5854 Used: 4878 Free: 976
 Nehalem    Total: 1255 Used: 1036 Free: 219
 Westmere   Total: 2285 Used: 2214 Free: 71

Currently queued jobs are requesting: 1463 Harpertown, 1767 Nehalem,
2860 Westmere nodes


Add "/u/scicon/tools/bin" to your PATH in .cshrc or other
shell start up scripts to avoid having to type in the complete
path for this tool.
```

You can also find for each job what processor models are used by looking at the "Model" field of the output of the command:

```
%qstat -a -W o=+model
```

- The Harpertown nodes in rack 32 have 16 GB memory/node instead of 8 GB/node.

  Example 2:

  This example shows a request of 2 nodes with bigmem in rack 32.

  ```
  #PBS -l select=2:ncpus=8:bigmem=true:model=har
  ```

- For a multi-node PBS job, the NCPUs used in each node can be different. This is useful for jobs that need more memory for some processes, but less for other processes. Resource requests can be done in "chunks" for a job with varying NCPUs per node.

  Example 3:

  This example shows a request of two chunks of resources. In the first chunk, 1 CPU in 1 node, and in the second chunk, 8 CPUs in each of three other nodes are requested:

  ```
  #PBS -l select=1:ncpus=1+3:ncpus=8
  ```

- A PBS job can run across different processor types. This can be useful in two scenarios:

    1. when you can not find enough free nodes within one model for your job
    2. when some of your processes need more memory while others need much less

    This can be accomplished by specifying the resources in "chunks", with one chunk asking for one processor type while another chunk asking for a different processor type.

    Example 4

    Here is an example to request 1 Westmere node (which provides 24 GB/node) and 3 Harpertown nodes (which provides 8 GB/node).

```
#PBS -lplace=scatter:excl:group=model
#PBS -lselect=1:ncpus=12:mpiprocs=12:model=wes+3:ncpus=8:mpiprocs=8:model=har
```

Default Variables Set by PBS

DRAFT

This article is being reviewed for completeness and technical accuracy.

You can use the "env" command--either in a PBS script or on the command line of a PBS interactive session--to find out what environment variables are set within a PBS job. In addition to the PBS_xxxx variables, the following two are useful to know:

- NCPUS defaults to number of CPUs that you requested for the node.
- OMP_NUM_THREADS defaults to 1 unless you explicitly set it to a different number.

  If your PBS job runs an OpenMP or MPI/OpenMP application, this variable sets the number of threads in the parallel region.
- FORT_BUFFERED defaults to 1.

  Setting this variable to 1 enables records to be accumulated in the buffer and flushed to disk later.

Sample PBS Script for Pleiades

DRAFT

This article is being reviewed for completeness and technical accuracy.

```
#PBS -S /bin/csh
#PBS -N cfd
# This example uses the Harpertown nodes
# User job can access ~7.6 GB of memory per Harpertown node.
# A memory intensive job that needs more than ~0.9 GB
# per process should use less than 8 cores per node
# to allow more memory per MPI process. This example
# asks for 64 nodes and 4 MPI processes per node.
# This request implies 64x4 = 256 MPI processes for the job.
#PBS -l select=64:ncpus=8:mpiprocs=4:model=har
#PBS -l walltime=4:00:00
#PBS -j oe
#PBS -W group_list=a0801
#PBS -m e

# Currently, there is no default compiler and MPI library set.
# You should load in the version you want.
# Currently, MVAPICH or SGI's MPT are available in 64-bit only,
# you should use a 64-bit version of the compiler.

module load comp-intel/11.1.046
module load mpi-sgi/mpt.2.04.10789

# By default, PBS executes your job from your home directory.
# However, you can use the environment variable
# PBS_O_WORKDIR to change to the directory where
# you submitted your job.

cd $PBS_O_WORKDIR

# use of dplace to pin processes to processors may improve performance
# Here you request to pin processes to processors 2, 3, 6, 7 of each node.
# This helps for using the Harpertown nodes, but not for Nehalem-EP or
# Westmere-EP nodes

# The resource request of select=64 and mpiprocs=4 implies
# that you want to have 256 MPI processes in total.
# If this is correct, you can omit the -np 256 for mpiexec
# that you might have used before.

mpiexec dplace -s1 -c2,3,6,7 ./grinder < run_input > output

# It is a good practice to write stderr and stdout to a file (ex: output)
# Otherwise, they will be written to the PBS stderr and stdout in /PBS/spool,
# which has limited amount  of space. When /PBS/spool is filled up, any job
# that tries to write to /PBS/spool will die.
```

```
# -end of script-
```

Pleiades devel Queue

NAS provides a special *devel* queue that provides faster turnaround when doing development work.

Currently, 512 Westmere nodes are reserved for the Pleiades *devel* queue, 24x7. The maximum wall-time allowed is 2:00:00 and the maximum NCPUS is 4800. Each user is allowed to have only one job running in the *devel* queue at any one time.

To improve PBS job scheduling response time, the *devel* queue and its resources (for example, nodes) have been moved to a second PBS server (pbspl3). With this move, users must specify the server name for jobs managed by pbspl3 with qsub, qstat, and qdel if the command is launched from a Pleiades front-end node (pfe[1-12] or bridge[1-4]). For example:

```
pfe1% qsub -q devel@pbspl3 job_script
1234.pbspl3.nas.nasa.gov

pfe1% qstat devel@pbspl3

pfe1% qstat -a @pbspl3

pfe1% qstat -u zsmith @pbspl3

pfe1% qstat 1234.pbspl3

pfe1% qdel 1234.pbspl3
```

Alternatively, if you set the environment variable **PBS_DEFAULT** to pbspl3, you can skip pbspl3 in your qsub, qstat, qdel commands. For example (in csh):

```
pfe1% setenv PBS_DEFAULT pbspl3

pfe1% qsub -q devel job_script
1234.pbspl3.nas.nasa.gov

pfe1% qstat devel

pfe1% qstat -a

pfe1% qstat -u zsmith

pfe1% qstat 1234

pfe1% qdel 1234
```

Use the csh command *unsetenv PBS_DEFAULT* to return to using the default PBS server, pbspl1.

Note that the changes described here do not apply to jobs submitted to the other queues

(normal, long, debug, and all special queues) served by the default server, pbspl1.

To see all jobs you have submitted to pbspl1 or pbspl3 (using username zsmith in the example below), type the following:

```
pfe1% qstat @pbspl1 @pbspl3 -W combine -u zsmith
```